2009-03

Team 6: Data Farming in Netcentric Systems Test Planning

Torres, Gil; Buscemi, Jim; Pickett, Kent; Hoivik, Tom; Sanchez, Susan; Upton, Stephen; Wan, Hong

https://hdl.handle.net/10945/35653

Team 6: Data Farming in Netcentric Systems Test Planning

TEAM 6 MEMBERS

Gil Torres
NAVAIR, US

Jim Buscemi
GBL Systems Corporation, US

Kent Pickett
The MITRE Corporation, US

Tom Hoivik
Susan Sanchez
Stephen Upton
Hong Wan
Naval Postgraduate School, US

INTRODUCTION

There is a need to conduct testing in a complex joint mission environment across the acquisition life cycle to improve a program manager's ability to deliver joint capabilities to warfighters. For a joint mission environment with many interdependent systems, assessing individual system and system-of-systems (SoS) contributions to joint mission effectiveness becomes extremely challenging. A change in one system may have cascading effects across the mission environment and, furthermore, many of these systems may be at different points in development and acquisition. This complex adaptive SoS environment makes it nearly impossible to plan efficient tests using current test methods and capabilities. Cogent planning for the tests of these complex adaptive systems involves a very tedious, almost impossible, test planning process for determining what and how exactly to test. To do this efficiently, new test and evaluation (T&E) tools, methods, and processes are needed and data farming has been identified as one tool that may help in this process.

For this workshop, our team continued work in exploring the use of data farming in the netcentric systems test planning process. Our objectives were to:

- Continue to gain a fuller understanding of the challenges in planning Netcentric Systems Tests;
- Continue to explore areas in Netcentric Systems Tests where data farming may be complementary to other techniques and tools;
- Continue to gain an appreciation for the state-of-the-art experimental design techniques and algorithms for exploring a large possibilities landscape; and,
- Test the capability of our prototype JTEAM (Joint Test and Evaluation Agent Model) framework in data farming a Joint Fires Scenario, focusing on the C2 system parameters. The scenario was developed to support the InterTEC (Interoperability Test and Evaluation Capability) Spiral 2 System Integration Test Plan (STIP).

To guide our discussion, as well as illustrate the data farming process, we conducted two notional experiments using a standard Nearly-Orthogonal Latin Hypercube (NOLH) design and a newer Resolution VII Fractional Factorial (R7FF) design. The system-of-systems under consideration in these experiments was based on a Time-Sensitive Target (TST) scenario vignette, which was partially implemented in JTEAM prior to the workshop, and a subset of the larger Joint Fires scenario. The next sections give a brief overview of the scenario, the JTEAM simulation, the experiments, and their resulting analyses. The article concludes with a summary and discussion of our future work.

SCENARIO

The scenario was a simplified variation of the TST vignette that was developed to support the InterTEC (Interoperability Test and Evaluation Capability) Spiral 2 System Integration Test Plan (STIP). On the Red side, the TST is a Red Convoy moving down a road. Protecting the convoy is a Red Air-Air (RedAA) aircraft, equipped with an Air-Air missile, providing defense against Blue's aircraft. On the Blue side,

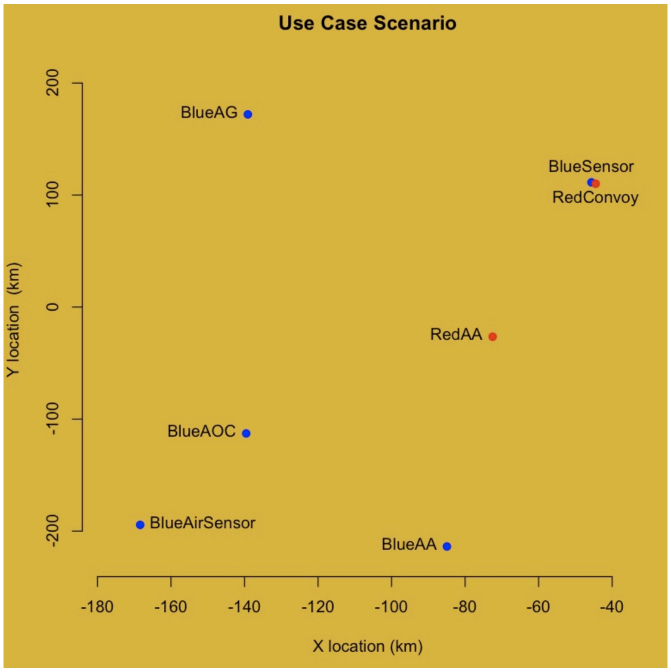

Figure 1: Use Case Scenario

there is a Blue Sensor (ground) that can detect and track the Red Convoy; a Blue Air-Air aircraft (BlueAA), equipped with a Blue Air-Air missile; a Blue Air-Ground (BlueAG) aircraft, equipped with a Blue Air-Ground missile; a Blue AirSensor that can detect the Red Air-Air aircraft; and a Blue AOC (Air Operations Center), which makes decisions on which resource to attack the TST. The scenario laydown is shown in Figure 1.

The scenario proceeds as follows: the Red Convoy begins moving down the road at simulation start; when within range of the Blue Sensor, the Blue Sensor sends a Call For Fire message to the Blue AOC; the Blue AOC decides which resource to send based on target priority, resource availability, weapon matching and fires area deconfliction, and sends a message to the first matching resource to conduct the mission. For this example, the only resource available is the BlueAG aircraft, which then proceeds to the target location indicated in the Call for Fire message. When nearing the TST (based on onboard sensor range), the BlueAG aircraft launches its AG missile and returns to its base.

Happening concurrently is a similar process on the Air-Air side. When the Blue AirSensor detects the RedAA, it determines which resource is available to attack that target (similar to the AG situation, the Blue AA is the only Anti-AA resource), and sends a message to the BlueAA to proceed to the target location. When the BlueAA nears the RedAA, it launches its AA missile. If the RedAA sensor's range is sufficient, it can also detect the Blue AA and launch its Red AA missile. Mission success is based on whether or not the Red Convoy is destroyed.

In the current implementation, the Air and Ground interactions are independent, e.g., the Blue AG can proceed to its target though the Red AA may still be a threat. Future work will focus on integrating these aspects.

JTEAM OVERVIEW

JTEAM (Joint Test and Evaluation Agent Model) is a prototype Agent-based simulation being developed as part of the JMEDF (Joint Mission Effectiveness support using Data Farming) project, supporting the Netcentric Systems Test Program. The goal of JTEAM is to help test designers in developing test designs for joint systems of systems tests by providing an easy-to-use, fast-running tool, and combined with state-of-the-art experimental design techniques, to explore a wide variety of test scenarios.

JTEAM is a discrete-event, three-dimensional, farmable agent-based model built on top of a composable and extensible framework. Farmability of the model enhances computational experiments by allowing users to easily vary input parameters associated with the agents. Composability allows users to build up or construct agents using software components specific to the domain. Extensibility allows users to develop their own software components to extend functionality provided by the basic framework.

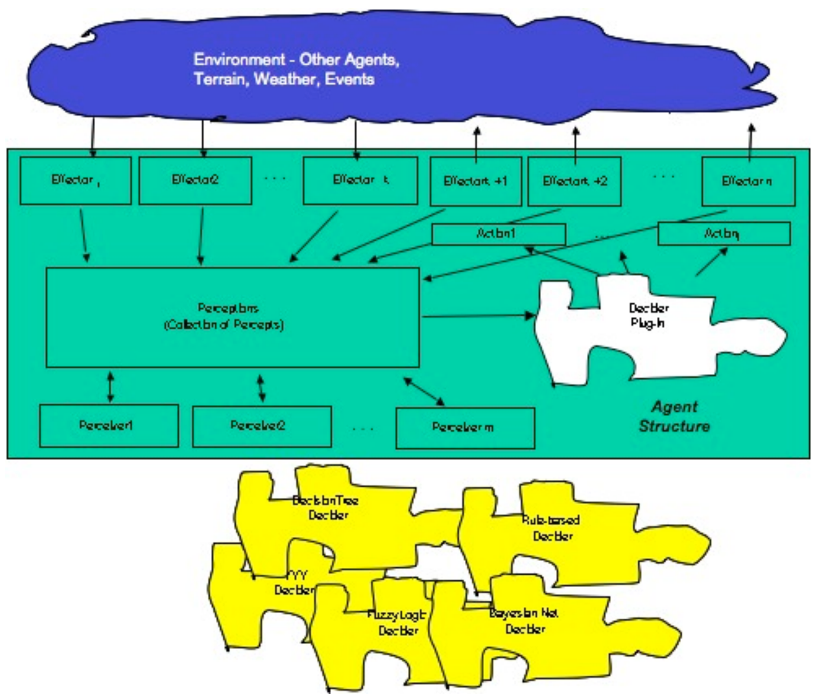

Figure 2: JTEAM Agent Structure

The JTEAM model is composed of a collection of agents and an underlying world model where the agents interact, which currently is a 3-dimensional spatial world with flat terrain. Each Agent has a basic structure that is common to all agents, with functionality added by including specific Decider, Effector and Perceiver components. In addition to these components, Agent's have a name, a side, an observableClass, and a targetClass, all of which can be set to arbitrary values. Finally, Agent's have a PropertyHandler that can handle a set of user-defined properties (through Effectors). The agents, and their components, are specified in an XML-formatted scenario file.

Each Decider, Effector, and Perceiver can have a set of farmable parameters associated with that component. Common structure includes communications, action, effector, damage, and perception handling mechanisms, and common properties such as target and observable class.

As depicted in Figure 2, each agent can have one or more Effectors, one or more Actions, one or more Perceivers, one Decider, and a Perceptions or "knowledge" base, which is a collection of Percepts that characterize the Agent's situational awareness.

Briefly, Effectors provide the Agent a means to observe or influence its external environment, through Actions, such as sensing, movement, or shooting. Effectors also provide Percepts, which are placed in the Perceptions base to be used by other Effectors or Perceivers. The Agent's set of Percepts constitutes what the Agent "knows" about the environment and itself. Perceivers work with Percepts in the Perceptions base, and provide additional, sometimes "higher-level" Percepts, or by filtering and removing Percepts to model such things as memory or operator overload. A Decider then uses those Percepts to "decide" on the set of Actions to take, and tasks the Effectors to carry out those Actions.

JTEAM is written in the Java language and uses the MASON agent-based modeling framework (specifically version 12) for its underlying simulation infrastructure, in addition to a number of other supporting open source packages that provide additional functionality.

Farming Parameters	Min	Max
Blue Sensor call for fire out process time	5 sec	30 sec
Blue Sensor comm link reliability	0.7	1
Blue Sensor sensor range	2 km	10 km
Blue Sensor probability of detection Red Convoy	0.7	1
Blue AOC call for fire in process time	5 sec	30 sec
Blue AOC comm link reliability	0.7	1
Blue AOC decision time	30 sec	2 min
Blue AOC range of the Blue AG resource	10 km	300 km
Blue AirSensor call for OCA out processing time	5 sec	30 sec
Blue AirSensor OCA out processing time	5 sec	30 sec
Blue AirSensor sensor range	200 km	400 km
Blue AirSensor probability of detection for Red AA	0.7	1
Blue AirSensor decision time for the mission	30 sec	2 min
Blue AirSensor range of the Blue AA resource	100 km	300 km
Blue AA OCA in process time	5 sec	30 sec
Blue AA speed	280 m/s	320 m/s
Blue AG JFIRES in process time	5 sec	30 sec
Blue AG speed	250 m/s	280 m/s
Red AA speed	280 m/s	320 m/s
Red Convoy speed	10 m/s	20 m/s
Blue AAM pk	0.8	1
Blue AAM speed	700 m/s	750 m/s
Red AAM pk	0.8	1
Red AAM speed	700 m/s	750 m/s
Blue AGM pk	0.8	1
Blue AGM speed	280 m/s	320 m/s

Table 1: Data Farming Parameters

DISCUSSION AND EXPERIMENTS

The group discussed several areas where data farming could potentially be useful in the Net-centric systems test planning process. The group also became more familiar with the NST planning process in order to understand where the challenges lie. The team used JTEAM with the TST scenario and focused on the associated C2 parameters, particularly time to make a decision as to what asset or platform to assign to the TST based on a priori known capabilities of the Blue agents.

Using a modified version of the initial TST scenario, the team developed a list of 65 potential factors that could be used in a design of experiments (DOE). In order to use the standard NOLH, in addition to one of the newer designs, we down selected from 65 factors to 26 factors. We then picked minimum and maximum values that seemed reasonable given the construct of the scenario. The parameters and their minimum and maximum values are listed in Table 1. We constructed a 26 factor NOLH of 257 design points, and ran JTEAM on the new SEED cluster, reaper, using 30 replications for each design point. We then conducted an initial analysis, demonstrating to the team members the types of analysis and information that can be obtained using primarily regression trees. Our primary MOE was the mean time to kill the convoy.

Following the NOLH runs, we generated a 26 factor, Resolution VII Fractional Factorial (R7FF) design, which resulted in 16384 design points. The R7FF is one of the new designs developed as part of the JMEDF project. We ran this experiment, again using reaper, with 5 replications for each design point.

RESULTS AND ANALYSIS

Our initial analysis using the 26 factor NOLH indicated that the AOC decision time was the most important factor, which was the factor the group "guessed right". Also, the mean time to kill was close to the "actual" 10 minutes, i.e., the time observed during the actual field test. However, while conducting the analysis using data from R7FF runs, we noticed that the initial NOLH results did not include output from the runs where the Red Convoy was not killed; the Red Convoy had survived 66% of the time. When the analysis was conducted by reweighing the effects of the non-kills, the AOC decision time was no longer a factor in any of the statistical models.

The results for the probability of kill for the TST ranged from completely ineffective to nearly perfect with just a few factors and splits of the regression tree, as can be seen in Figure 3. Similar results were obtained for the mean Time To Kill MOE. It appears that further work will be needed in JTEAM post-processing tools to capture the relevant output data more effectively!

FUTURE WORK

One of our objectives for the workshop was to arrive at a way forward for the project. After discussion and seeing what data farming could do, the group decided on several items to accomplish by the end of year review, which is to occur this September. The tasks that lie ahead include:

1. Implementing a sequential design, such as the R7FF, so that it could be used by lay persons.
2. Implement a JTEAM Decider component that would focus on modeling some aspect of the AOC.
3. Expand the scenario to incorporate other agents and complex decision-making for the AOC agent.
4. Include the Decider component as a factor in the upcoming analyses.

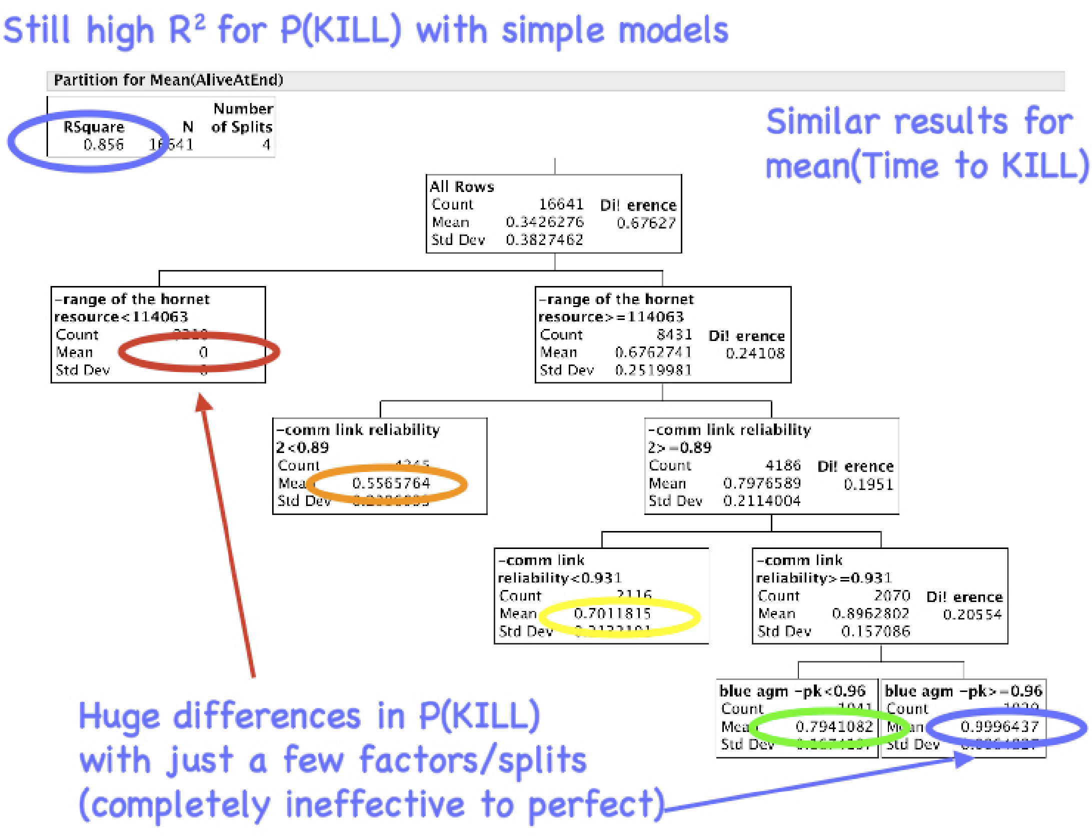

Figure 3- Partition tree for P(Kill)

SUMMARY

To accomplish our workshop objectives, our team conducted two notional data farming experiments in order to gain a better understanding of the potential applications of the data farming process, techniques and tools to the test planning process. We used a TST scenario, implemented in our prototype JTEAM model, and made over 89,000 runs, using both a standard NOLH design and one of the newer R7FF designs. We conducted an analysis of that data, illustrating several standard analysis products. Finally, we discussed the way forward for our project's end of year review.